Thoughts on modularization

- Structuring JavaScript code - why?
- The technical basis: IIFEs and the Module Pattern
- CommonJS Modules
- AMD, the Asynchronous Module Definition
- Require.js
- Combining loader techniques

Structuring JavaScript Code - Why?

- Writing applications server-side in JavaScript
- Client applications on HTML/JS basis
- Mobile apps
- Reusable modules

Iffy?

- The Module Pattern - mostly attributed to Douglas Crockford
- Based on IIFEs - Immediately Invoked Function Expressions

Demo

The Module Pattern

Debugging

- Common criticism about the Module Pattern: hard to debug
- Be disciplined with your public APIs!

Demo

Debugging the Module Pattern

“Standard” module systems

- CommonJS modules
  - Mainly used server side
- AMD, the Asynchronous Module Definition
  - Mainly used in the browser
- [ECMA Script 6 has a module system built in that combines the best of both worlds and a bit more.]

CommonJS Modules

- Use ‘exports’ object in modules to “publish” API
- ‘require’ function used to load module
- ‘require’ must return immediately, async loading takes some effort

Demo

CommonJS Modules

AMD Modules

- A ‘define’ function is called with the module name, dependencies and a callback
- A module loader resolves the dependencies when a ‘require’ call is made

Demo

AMD Modules

If you're creating modules...

- Consider making your module available under both schemes

More things to check out

- RequireJS
  - Special support for jQuery (and sub-module) loading
  - Optimizer - combine and minify
  - Use in Node.js for AMD loading server-side
- Browserify & Co - use CommonJS modules in the browser automatically